Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Кафедра информатики
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____________ А.С. Кузнецов
« 09 » 06 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

09.03.04 – Программная инженерия
код – наименование направления

Программное средство для исследования свойств двух алгоритмов глобальной оптимизации
тема

Руководитель			09.06.17 подпись, дата	А.С. Михалев инициалы, фамилия
Выпускник			09.06.17 подпись, дата	А.В. Резин инициалы, фамилия
Консультант	профессор,	д.т.н	09.06.17 подпись, дата	А.И. Рубан инициалы, фамилия
Нормконтролер	доцент,	к.т.н.	09.06.17 подпись, дата	О.А. Антамошкин инициалы, фамилия

Красноярск 2017

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Кафедра информатики
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
__________ А.С. Кузнецов
«18» 05 2017 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАФИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту: Резину Андрею Владимировичу

Группа: КИ13-18Б. Направление (специальность): 09.03.04 Программная инженерия.

Тема выпускной квалификационной работы: «Программное средство для исследования свойств двух алгоритмов глобальной оптимизации».

Утверждена приказом по университету №2930 от 07 марта 2017 г.

Руководитель: А.С. Михалев, ассистент кафедры «Информатика»;

Консультант: А.И. Рубан, доктор технических наук, профессор кафедры «Информатика».

Исходные данные для ВКР: проектирование, разработка и реализация программного средства для исследования свойств двух алгоритмов глобальной оптимизации.

Перечень основных тем ВКР:

- Введение;
- Глава 1. Общие положения;
- Глава 2. Генетический алгоритм;
- Глава 3. Алгоритм, основанный на селективном усреднении координат;
- Глава 4. Описание программного средства;
- Глава 5. Численные исследования;
- ЗАКЛЮЧЕНИЕ.

Перечень графического или иллюстрационного материала с указанием основных чертежей, плакатов, слайдов: презентационные слайды PowerPoint.

Руководитель ______________ А.С. Михалев

(подпись)

Консультант ______________ А.И. Рубан

(подпись)

Задание принял к исполнению ______________ А.В. Резин

(подпись)

«18» 05 2017 г.

РЕФЕРАТ

Бакалаврская работа 46 стр., 20 рисунок, 6 источников.

Объект исследования – алгоритмы глобальной оптимизации.

Цель работы – разработать универсальное программное средство для исследования свойств алгоритмов глобальной оптимизации.

Метод исследования – практический эксперимент.

Результат – изучена предметная область по данной теме, реализованы два алгоритма глобальной оптимизации, реализовано программное средство исследования алгоритмов глобальной оптимизации.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Задача поиска оптимального значения целевой функции является одной из актуальных в области науки и техники. Большинство практических задач, возникающих в различных сферах человеческой деятельности, могут быть сведены к задачам оптимизации. Это различные задачи расчета параметров конструкций, траектории движения объекта, распределения ресурсов. Применение стратегии поиска оптимума для построения и решения задач синтеза сложных систем приводит к повышению качества управления, планирования и проектирования.

При быстро возрастающей сложности оптимизируемых объектов совершенно естественным является и усложнение их математических моделей, что, как следствие, значительно затрудняет поиск оптимальной комбинации параметров. Очень часто не представляется возможным найти такую комбинацию аналитически и возникает необходимость построения численных методов для ее поиска.

Проблема численного решения задач оптимизации, в свою очередь, сопряжена со значительными трудностями. Во многом они связаны с их размерностью, видом оптимизирующей целевой функции и наличием помех. Такие задачи характеризуются целевой функцией со следующими свойствами. Во-первых, она может быть многоэкстремальной, недифференцируемой и, более того, заданной в форме черного ящика (т. е. в виде некоторой вычислительной процедуры или прибора, на вход которого подается аргумент, а на выходе наблюдается соответствующее значение функции). Во-вторых, каждое вычисление функции в некоторой точке допустимой области может требовать значительных вычислительных ресурсов.

Увеличение числа прикладных проблем, описываемых математическими моделями подобного типа, и бурное развитие вычислительной техники вызвали значительный интерес к указанным задачам и инициировали развитие глобальной оптимизации как области математического программирования, занимающейся разработкой теории и методов решения многоэкстремальных оптимизационных задач.

Численные методы глобальной оптимизации существенно отличаются от стандартных локальных методов поиска, которые часто неспособны найти глобальное (т. е. абсолютно лучшее) решение рассматриваемых задач в силу многоэкстремальности целевой функции. Локальные методы, как правило, оказываются не в состоянии покинуть зоны притяжения локальных оптимумов и, соответственно, упускают глобальный оптимум. Использование же найденных локальных решений может оказаться недостаточным, поскольку глобальное решение может дать существенный выигрыш по сравнению с локальными.

Как уже было сказано, особенностями решаемых задач оптимизации являются нелинейность, недифференцируемость, многоэкстремальность (мультимодальность), овражность, отсутствие аналитического выражения (плохая формализованность) и высокая вычислительная сложность оптимизируемых функций, высокая размерность пространства поиска, сложная топология области допустимых значений. На сегодняшний день для их решения получено чрезвычайно большое число алгоритмов глобальной оптимизации. В свою очередь среди них нет универсального метода для решения задач. В частности, решение двух различных задач глобальной оптимизации выбранным алгоритмом в первом случае (для первой задачи) даст оптимальное решение, во втором (для второй задачи) – оптимальное решение не будет достигнуто. В связи с этим актуальной становится разработка эффективных и достаточно универсальных алгоритмов глобальной оптимизации.

В свою очередь не престает быть актуальной задача исследования уже существующих и зарекомендовавших себя алгоритмов глобальной оптимизации.

Для каждой задачи применение алгоритмов глобальной оптимизации требует настройки соответствующих параметров и нет универсального средства для решения поставленной задачи, учитывая, что задача глобальной оптимизации актуальна на сегодняшний день можно сделать вывод, что необходимо программное средство наглядно указывающая достоинства и недостатки одного из выбранных алгоритмов.

Данные алгоритмы имеют ряд преимуществ и недостатков друг относительно друга и реализуемое программное средство призвано на практике их показать.

Цель работы. Проектирование и реализация универсального программного средства для исследования свойств алгоритмов глобальной оптимизации. Исследование свойств двух выбранных алгоритмов: генетического алгоритма и алгоритма, основанного на селективном усреднении переменных на тестовых задачах.

Постановка задачи. Разработать универсальное программное средство для решения задач оптимизации и исследования свойств алгоритмов.

Объём и структура работы. Выпускная квалификационная работа состоит из введения, 5 глав и заключения. Общий объем выпускной квалификационной работы 46 страниц, включая библиографический список из 6 источников. Иллюстративный материал представлен на 20 рисунке.

1. Общие положения

1.1 Постановка задачи глобальной оптимизации

В общей постановке задача глобальной оптимизации формулируется следующим образом [1]:

$$f^* = f(X^*) = \min_{X \in R^m} f(X),$$

где $f(x)$ – целевая функция,

x^* – глобально-оптимальная точка или глобально-оптимальное решение,

$X = (x_1, \dots, x_m)$ – m-мерный вектор искомых переменных,

R^m – m-мерное арифметическое пространство (Евклидово пространство).

Область, в которой отыскивается решение задачи глобальной оптимизации, задается следующим образом:

$$D = \{X | x_i^{min} \le x_i \le x_i^{max}, i = \overline{1, m}\} \subset R^m.$$

Функция ограничена и непрерывна почти всюду на X. Область X состоит из конечного или счетного множества замкнутых ограниченных не пересекающихся подобластей, в каждой из которых функция $f(X)$ ограничена и непрерывна.

Необходимо в X найти глобальный экстремум функции $f(X)$ при условии его единственности. Глобальный экстремум может находиться внутри области X, на ее границе, а также на границах подобластей, из которых она состоит.

1.2 Методы решения задач глобальной оптимизации

Задача глобальной оптимизации в силу широты класса многоэкстремальных функций в общем случае является неразрешимой [2], т.е. нельзя гарантировать, что решение задачи будет получено за конечное число шагов. Таким образом, специфика задачи глобальной оптимизации заключается в многоэкстремальности функции цели и неразрешимости в общем случае.

Существует два направления в работе с неразрешимыми задачами. Первый способ заключается в использовании априорной информации о целевой функции и множестве оптимизации, которая приводит исходную задачу к разрешимому виду или, по крайней мере, дает возможность с уверенностью говорить о том, что точное решение будет найдено. Второй подход позволяет рассматривать более широкий класс функций цели: отказываясь от требования разрешимости задачи, необходимо получить оценку глобального решения. При таком подходе желательно иметь также некоторый критерий приемлемости полученной оценки.

Сложность задачи оптимизации определяется свойствами функции f и множества X. Имеет место определенная дуальность свойств области X и функции f. Задачу со сложной целевой функцией можно переформулировать так, что целевая функция станет простой, но при этом все сложности будут перенесены на область оптимизации. Однако чаще всего встречаются задачи с достаточно простыми областями X, а все сложности переносятся на целевую функцию. Границы множества определяют априорную информацию о положении x^*: чем шире границы X, тем больше априорная неопределенность о положении x^* и тем труднее поиск этой точки. Предположение о замкнутости X вместе с обязательным предположением о непрерывности функции f в окрестности точки глобального минимума гарантируют, что $x^* \in X$, т.е. что глобальный минимум f в X достигается.

Если множество X задается с помощью различных ограничений типа равенств и неравенств и имеет сложную структуру, задача глобальной оптимизации может быть сведена к задаче оптимизации на множестве более простой структуры.

Основной задачей глобальной является выявление тенденций глобального поведения целевой функции. Экстремум целевой функции приходится искаться в условиях нелинейности ее и ограничений на параметры, недоступности или отсутствия информации об объекте исследования. Процесс нахождения глобального минимума является итерационным, что порождает последовательность точек в соответствии с предписанным набором правил, включающий критерий останова счета. Глобальное решение задачи оптимизации заключается в полном переборе всех ее локальные решения. Такая задача, как правило, оказывается трудоемкой. Другой подход – перебрать часть локальных решений и показать, что оставшиеся локальные минимумы не влияют на точность решения. Таким образом, идея большинства методов глобальной оптимизации – оценить значения целевой функции $f(\cdot)$ на некотором множестве точек из допустимого множества X. Различие заключается в способах выбора этих точек.

Другим важным свойством алгоритмов оптимизации является сходимость генерируемой им последовательности точек к глобальному оптимальному решению. Однако в большинстве случаев получаются менее благоприятные результаты: невыпуклость функций, большая размерность задачи или другие трудности вынуждают останавливать алгоритм, если получена точка, принадлежащая некоторому множеству приближенных решений. Другими словами, для любого численного алгоритма необходимы условия остановки.

1.3 Классификация методов глобальной оптимизации

Основным стержнем структурной классификации алгоритмов глобальной оптимизации является деление их на детерминированные, стохастические и комбинированные.

Детерминированные методы получают глобальное решение посредством исчерпывающего поиска на всем допустимом множестве [3].

Вследствие этого, большинство детерминированных методов теряют эффективность и надежность с возрастанием размерности задачи.

Стохастические алгоритмы позволяют уйти от проблем детерминированных алгоритмов. Одним из основных принципов организации стохастических методов является оценивание значения функции цели в случайных точках допустимого множества с последующей обработкой получившейся выборки. Однако стохастические методы не гарантируют, что будет найден именно глобальный оптимум [4].

Таким образом, можно сделать следующее заключение:

1. Стохастический алгоритм никогда не дает стопроцентной гарантии нахождения глобального минимума, поскольку такая гарантия требует бесконечного времени выполнения.

2. Детерминированные алгоритмы всегда выполняются за конечное (хотя зачастую достаточно длительное) время и возвращают множество областей, гарантированно содержащее все глобальные минимумы.

3. Стохастические алгоритмы не гарантируют нахождения всех глобальных минимумов функции, они находят только один глобальный минимум, в то время как может существовать несколько альтернативных глобальных минимумов (задача поиска главных минимумов). Детерминированные алгоритмы дают гарантию того, что все глобальные минимумы содержатся в областях, полученных после завершения алгоритма.

Несмотря на это, среди глобальных методов оптимизации все большую популярность приобретают алгоритмы стохастической оптимизации, имеющие слабую доказательную базу, но зачастую, демонстрирующие прекрасные результаты при решении практических задач. Связана данная популярность с отсутствием априорной информации, в большинстве инженерных задач, о характере глобального поведения целевой функции, совмещенная с её сильной нелинейностью и зависимостью от множества непрерывных переменных.

Другой эффективный подход к решению задач глобальной оптимизации в настоящее время заключается в разработке комбинированных существующих алгоритмов. В гибридных (комбинированных) алгоритмах, объединяющих различные либо одинаковые алгоритмы, но с различными значениями параметров, преимущества одного алгоритма могут компенсировать недостатки другого.

В данной работе для исследования было выбрано два алгоритма. Первый – генетический – достаточно широко используемый в задачах глобальной оптимизации стохастический алгоритм. Второй – алгоритм, основанный на селективном усреднении искомых переменных – сочетающий в себе детерминированный метод поиска со случайным просмотром области поиска.

2. Генетический алгоритм

Генетический алгоритм – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе [5].

Генетический алгоритм является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе. Общая схема работы любого генетического алгоритма представлена на рисунке 1.

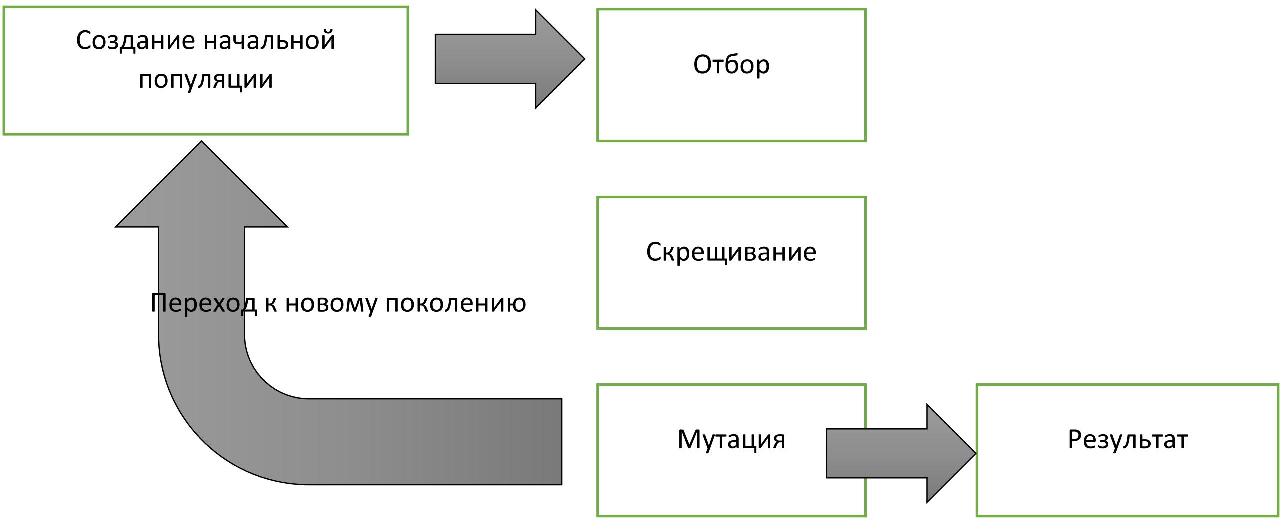

Рисунок 1 – Схема работы генетического алгоритма

Работа алгоритма начинается с генерации начальной популяции $S^0 = \{s_1{}^0, \dots, s_n{}^0\}$, $s_i{}^0 \in X$, состоящей из n хромосом (генотипов). Далее, каждая хромосома оцениваются с использованием «функции приспособленности», в результате чего с каждой хромосомой ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо фенотип (смысловое содержание генотипа), им описываемый, решает поставленную задачу.

Оценка приспособленности проводится в две стадии:

1. Расчет «функции приспособленности»

$$f^k = \{{f_1}^k \dots {f_n}^k\}, {f_i}^k = W({s_i}^k)$$

2. Дополнительное преобразование, например, нормировка

$$f^{k'} = \left\{{f_1}^{k'} \dots {f_n}^{k'}\right\}, {f_i}^{k'} = \frac{{f_i}^k - f_0}{f_1 - f_0},$$

где f_1 – лучший показатель в текущей популяции;

f_0 – худший показатель в текущей популяции.

Из полученного множества решений с учетом значения «приспособленности» выбираются решения (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» - отбор, скрещивание и мутация, результатом чего является получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение.

Этот набор действий повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма. Таким критерием может быть:

- схождение популяции;
- количество поколений (циклов) достигнет заранее выбранного максимума;
- исчерпание времени, отпущенного на эволюцию.

Схождением называется такое состояние популяции, когда все строки популяции почти одинаковы и находятся в области некоторого экстремума. В такой ситуации кроссинговер практически никак не изменяет популяции, т.к. создаваемые при нем потомки представляют собой копии родителей с переменными участками хромосом. Вышедшие из этой области за счет мутации особи склонны вымирать, т.к. чаще имеют меньшую приспособленность, особенно если данный экстремум является глобальным максимумом. Схождение популяций обычно означает, что найдено лучшее или близкое к нему решение.

Таким образом, можно выделить следующие этапы генетического алгоритма:

1. Задать целевую функцию (приспособленности) для особей популяции.
2. Создать начальную популяцию.
3. Скрещивание.
4. Мутация.
5. Вычислить значение целевой функции для всех особей.
6. Формирование нового поколения (селекция).
7. Если выполняются условия останова, то конец, иначе – переход в шагу 3.

2.1 Перечень настраиваемых параметров

Настраиваемыми параметрами генетического алгоритма являются методы селекции, скрещивания, формирования нового поколения и выбора родителей, а также размер популяции.

2.1.1 Отбор (селекция)

На этапе отбора нужно из всей популяции выбрать определённую её долю, которая останется «в живых» на данном этапе эволюции. Есть разные способы проводить отбор. Вероятность выживания особи H должна зависеть от значения функции приспособленности. Сама доля выживших S обычно является параметром генетического алгоритма, и её просто задают заранее. По итогам отбора из N особей популяции H должны остаться S_N особей, которые войдут в итоговую популяцию H'. Остальные особи погибают.

В данной работе для отбора особей используются следующие подходы:

- Турнирная селекция,
- Метод рулетки,
- Ранговая селекция.

Турнирная селекция. При турнирной селекции все особи популяции разбиваются на группы с выбором в каждой из них особи/особей с наилучшей приспособленностью. Подгруппы могут иметь произвольный размер, но при классическом подходе популяция делится на подгруппы размером по 2-3 особи.

Различаются два способа такого выбора: детерминированный выбор и случайный выбор. Детерминированный выбор осуществляется с вероятностью, равной 1, а случайный выбор – с вероятностью меньшей 1.

Схема, которая иллюстрирует метод турнирной селекции для подгрупп, состоящих из двух особей, представлена на рисунке 2.

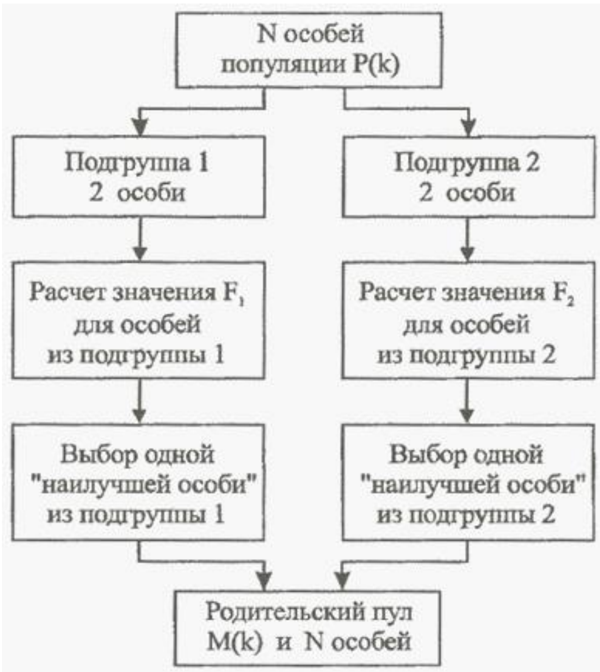

Рисунок 2 – Принцип турнирной селекции

Метод рулетки. При этом методе отбора хромосомы - кандидаты из t-го поколения $S(t)$ выбираются для выживания в следующем, (t+1)-м поколении S(t+1) путем использования колеса рулетки, где каждая хромосома s^t в популяции представлена на колесе в виде сектора, ширина которого пропорциональна соответствующему значению «функции приспособленности»:

$$P(i) = \frac{f(i)}{\sum_{i=1}^{N} f(i)}$$

Таким образом, те хромосомы, которые имеют большую пригодность, соответствуют большему сектору на колесе, а хромосомы с меньшей пригодностью - сравнительно малому сектору на колесе рулетки. Процедура отбора сводится к вращению колеса рулетки M раз и принятием в качестве кандидатов в следующем поколении тех хромосом ${s_1}^t, \dots, {s_n}^t$, которые будут выделены по завершении вращения. При таком отборе члены популяции с более высокой приспособленностью с большей вероятностью будут чаще выбираться, чем особи с низкой приспособленностью. Пример рулеточной селекции представлен на рисунке 3.

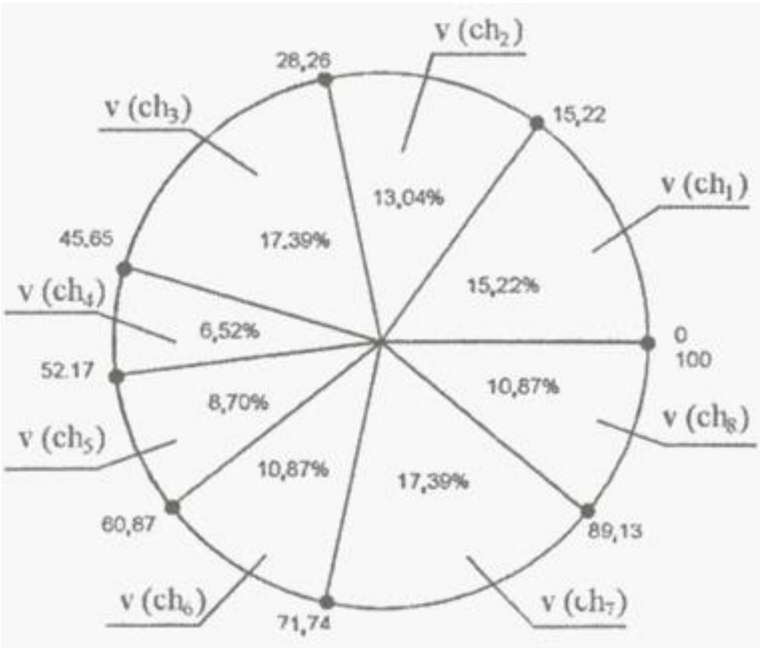

Рисунок 3 – Принцип рулеточной селекции

Ранговая селекция. При ранговой селекции особи популяции ранжируются по значениям их функции приспособленности. Это можно представить как отсортированный список особей, упорядоченных по направлению от наиболее приспособленных к наименее приспособленным (или наоборот), в котором каждой особи приписывается число, определяющее ее место в списке, и называемое рангом. Количество копий $M(k)$ каждой особи, введенных в родительскую популяцию, рассчитывается по априорно заданной функции в зависимости от ранга особи. Пример такой функции представлен на рисунке 4.

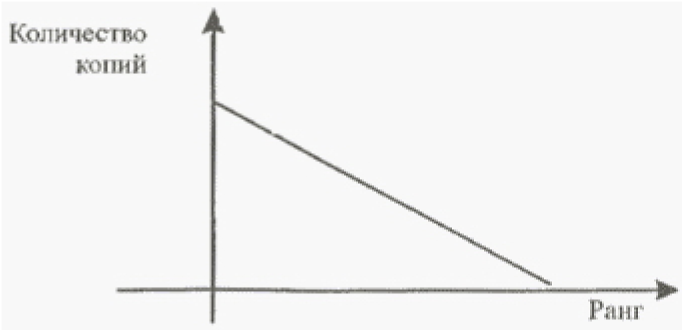

Рисунок 4 – Принцип ранговой селекции

2.1.2 Выбор родителей

Размножение в генетических алгоритмах обычно половое – чтобы произвести потомка, нужны несколько родителей, обычно два. Можно выделить несколько операторов выбора родителей: панмиксия, инбридинг и аутбридинг.

Панмиксия. Оба родителя выбираются случайно, каждая особь популяции имеет равные шансы быть выбранной.

Инбридинг. Первый родитель выбирается случайно, а вторым выбирается такой, который наиболее похож на первого родителя.

Аутбридинг. Первый родитель выбирается случайно, а вторым выбирается такой, который наиболее не похож на первого родителя.

2.1.3 Размножение (скрещивание)

Оператор скрещивания представляет собой процесс формирования двух новых потомков для двух выбранных хромосом. В разных алгоритмах размножение определяется по-разному – зависит оно от представления данных. Главное требование к размножению — чтобы потомок или потомки имели возможность унаследовать черты обоих родителей, «смешав» их каким-либо способом.

Особенность размножения в генетических алгоритмах заключается в том, особи для размножения обычно выбираются из всей популяции H, а не из выживших на первом шаге элементов H'. Объясняется это тем, что среди особей нет разнообразия. Достаточно быстро среди особей появляется генотип, соответствующий локальному оптимуму, в результате чего все оставшиеся особи в популяции проигрывают ему в отборе, в результате чего вся популяция состоит из копий этой особи.

Есть разные способы борьбы с таким нежелательным эффектом; один из них – выбор для размножения не самых приспособленных, но вообще всех особей. Однако такой подход вынуждает хранить всех существовавших ранее особей, что увеличивает вычислительную сложность задачи. Поэтому часто применяют методы отбора особей для скрещивания таким образом, чтобы «размножались» не только самые приспособленные, но и другие особи, обладающие плохой приспособленностью.

Одноточечный кроссовер

Одноточечный кроссовер работает следующим образом. Пусть, например, имеются две родительские особи с хромосомами:

$$X = \{x_i, i \in [0; L]\} \text{и } Y = \{y_i, i \in [0, L]\}$$

Случайным образом определяется точка внутри хромосомы (точка разрыва), в которой обе хромосомы делятся на две части и обмениваются ими:

$$x_1 x_2 x_3 \dots x_{n-1} | x_n \dots x_m \to x_1 x_2 x_3 \dots x_{n-1} y_n \cdots y_m$$

$$y_1 y_2 y_3 \cdots y_{n-1} | y_n \cdots y_m \to y_1 y_2 y_3 \cdots y_{n-1} x_n \cdots x_m$$

Многоточечный кроссовер

При двухточечном кроссовере выбираются две точки разрыва, и родительские хромосомы обмениваются сегментом, который находится между двумя этими точками, например:

$$x_1 x_2 x_3 | x_4 \dots x_{n-1} | x_n \dots x_m \to x_1 x_2 x_3 y_4 \dots y_{n-1} x_n \dots x_m$$

$$y_1y_2y_3|y_4 \cdots y_{n-1}|y_n \cdots y_m \rightarrow y_1y_2y_3x_4 \cdots x_{n-1}y_n \cdots y_m$$

Равномерный кроссовер

При равномерном кроссовере каждый бит первого родителя наследуется первым потомком с заданной вероятностью; в противном случае этот бит передается второму потомку.

2.1.4 Стратегия формирования нового поколения

«Элитарный» способ

При данном подходе часть родительской популяции гарантированно переходит в новое поколение. После оценки текущей популяции выбираются несколько наилучших особей, они "автоматически" переходят в следующее поколение. При этом «элитные» особи могут принимать участие в скрещивании также, как и все остальные особи. При этом подходе родительские особи не соревнуются с потомками, поэтому подход получил название неконкурентный.

Классическая стратегия

Классическая стратегия формирования нового поколения включает два следующих подхода:

1) Новые особи (потомки) занимают места родителей. После чего начинается новый этап, в котором потомки оцениваются, выбираются, дают потомство и уступают место своим «детям».

2) Создается промежуточная популяция, которая включает в себя «родителей» и «детей». Члены этой популяции оцениваются, после этого из них выбираются N лучших, которые войдут в следующее поколение.

3 Алгоритм, основанный на селективном усреднении искомых переменных

Среди существующих методов поиска глобального экстремума другими перспективными подходами к решению задач являются рандомизированные методы оптимизации. Они чаще всего основаны на сочетании детерминированных методов поиска со случайным просмотром области просмотра. Суть этих алгоритмов в следующем:

- используя рандомизированный подход, проводится этап обучения (пробные движения) – вычисление значения функции качества в n точках;
- на основании предыдущего этапа рассчитывается направление движения к минимуму;
- вдоль полученного направления делается рабочее перемещение к экстремуму;
- осуществляется переход от минимизации исходной функции к минимизации новой усредненной функции, имеющей один экстремум в точке глобального минимума исходной функции;
- после коррекции размеров исследуемой области и совершения рабочего шага алгоритм повторяется до выполнения условия останова алгоритма.

При решении задач глобальной оптимизации среди рандомизированных подходов перспективными является метод, основанный на усреднении искомых переменных [6]. В основе метода селективного усреднения лежит разнесение во времени пробных и рабочих шагов. Перед совершением каждого рабочего шага производится серия пробных перемещений: в допустимой области равномерно размещаются пробные точки $x^{(i)}, i = \overline{1, n}$. За счет охвата пробными точками всей области поиска экстремума на рабочем шаге удается получить глобальные характеристики и организовать более надежное движение к экстремуму.

Для получения пробных точек $x^{(i)}$ последовательно генерируются точки в прямоугольной области Π^l с центром в точке x^l:

$$x_v^{(i)} = x_v^l + \Delta x_v^l u_v^{(i)}, u_v \in [-1; 1], v = \overline{1, m}, i = 1,2, \dots$$

В полученных пробных точках осуществляется измерение (или вычисление) оптимизируемой функции $f^{(i)} = f\left(x^{(i)}\right), i = \overline{1, n}.$ Далее алгоритм переходит к выполнению рабочего шага.

На каждом рабочем шаге выполняется операция селективного усреднения искомых переменных по результатам экспериментальных данных, полученных в пробных точках. За счет этого осуществляется переход в новую точку, в среднем более близкую к глобальному минимуму и адаптивная перестройка размеров прямоугольной области пробных движений. Новое значение x_v^{l+1}, в среднем более близкое к положению глобального минимума, и размеры прямоугольной области поисковых движений Δx_v^{l+1} вычисляются по следующим формулам:

$$x_v^{l+1} = x_v^l + \Delta x_v^l u_{v,min}, u_{v,min} = \sum_{i=1}^{n} u_v^{(i)} \bar{p}_{s,min}^{(i)}, v = \overline{1, k},$$

$$\bar{p}_{s,min}^{(i)} = \frac{p_s(g_{min}^{(i)})}{\sum_{j=1}^{n} p_s(g_{min}^{(j)})}, g_{min}^{(i)} = \frac{f^{(i)} - \hat{f}_{min}}{\hat{f}_{max} - \hat{f}_{min}},$$

$$\Delta x_v^{l+1} = \gamma_q \Delta x_v^l \left(\sum_{i=1}^{n} \left|u_v^{(i)}\right|^q \bar{p}_{s,min}^{(i)}\right)^{1/q},$$

$$l = 0,1,2, \dots; [\gamma_q > 0, q \in \{1,2, \dots\}, s > 0],$$

где, $\hat{f}_{max} = \max\{f^{(i)}, i = \overline{1, n}\}$,

$\hat{f}_{min} = \min\{f^{(i)}, i = \overline{1, n}\}$,

$p_s(*)$ – ядро,

$\bar{p}_{s,min}(*)$ – нормированное ядро,

s – степень селективности.

После перехода в новую точку все вышеуказанные действия повторяются до выполнения условия(ий) останова алгоритма. Останов алгоритма можно проводить либо по величине сжатия области пробных движений

$$\max\{\Delta x_v^l, v = \overline{1, m}\} \leq \varepsilon_1$$

либо по величине наибольшего уклонения минимизируемой функции на множестве пробных точек

$$|\hat{f}_{max} - \hat{f}_{min}| \leq \varepsilon_2.$$

Таким образом, можно выделить следующие этапы алгоритма, основанного на селективном усреднении искомых переменных:

1. Генерация пробных точек в прямоугольной области,
2. Расчет значений целевой функции в пробных точках,
3. Перевод значений целевой функции в относительные величины,
4. Расчет значений ядер для значений целевой функции,
5. Расчет значений нормированных ядер,
6. Расчет нового оптимального значения,
7. Расчет значений интервалов варьирования (размеров прямоугольной области поисковых движений).

3.1 Перечень настраиваемых параметров алгоритма

Настраиваемыми параметрами алгоритма, основанного на селективном усреднении искомых переменных являются количество пробных точек, вид ядра и степень селективности.

Основные свойства алгоритма зависят от типа ядра и его селектирующей способности. Выбор вида ядра и параметра селективности позволяет повышать скорость сжатия области поисковых движений. Но в то же время она должна быть такой, чтобы не пропустить глобальный экстремум. В свою очередь при увеличении s растет «селективность» (способность к локализации положения глобального экстремума) нормированного ядра $\bar{p}_{s,min}(x)$.

3.1.1 Ядерные функции

Для исследований в данной работе используются следующие виды ядер:

- Экспоненциальное в степени s

$$p_s(g) = \exp(-sg)$$

- Гиперболическое в степени s

$$p_s(g) = g^{-s}$$

- Линейное в степени s

$$p_s(g) = (1 - g)^s$$

- Параболическое в степени s

$$p_s(g) = (1 - g^2)^s$$

- Кубическое в степени s

$$p_s(g) = (1 - g^3)^s$$

4 Описание программной реализации

4.1 Обзор существующих аналогов

Алгоритмы глобальной оптимизации относится к классу алгоритмов, требующих больших вычислительных мощностей, поскольку он основывается на многократных вычислениях функции качества и ограничений в пробных точках. В связи с этим появляется необходимость выбора какого–либо инструментального средства для проведения исследований. В данной ситуации возможны два подхода. Первый заключается в использовании универсальных математических пакетов, второй – самостоятельное проектирование и разработка программного средства исследований.

На сегодняшний день создано весьма много коммерческих программных продуктов, реализующих тот или иной (а то и несколько различных), метод оптимизации. Наиболее известные из них: Matlab, Mathematica.

Остановимся для анализа на системе Matlab. Matlab – это высокоуровневый язык и интерактивная среда для программирования, численных расчетов и визуализации результатов. С помощью MATLAB можно анализировать данные, разрабатывать алгоритмы, создавать модели и приложения.

В случае использования уже готовых программных продуктов можно выделить следующие преимущества и недостатки.

Преимущества:

1. Готовый программный продукт уже готов к непосредственному использованию, что позволяет сразу приступить к исследованиям, а не тратить время на разработку и программирование собственных средств исследований.
2. Математические пакеты разрабатываются фирмами, которые работают в этом направлении уже долгое время и имеют всеобщее признание, что гарантирует качество таких продуктов. По этой

причине в них присутствуют широкие возможности для визуализации результатов исследований и обработки статистических данных, которые обязательно будут накапливаться в процессе проведения исследований. По причине популярности таких пакетов, в них решена проблема переноса под различные аппаратные платформы и операционные системы.

Недостатки:

1. Математические пакеты имеют высокую цену, чтобы можно было их использовать.
2. В широко распространенных математических пакетах методы оптимизации скрыты от пользователя. Это не позволяет наглядно представить процесс оптимизации, исследовать функции, управлять параметрами алгоритмов.
3. Математический пакет может не располагать всем необходимым для данных исследований набором сервисных утилит, в силу оригинальности самого алгоритма.

Именно по причине нестандартности решаемой задачи был сделан выбор в пользу разработки собственного программного средства для проведения исследований, в котором можно было бы реализовать весь необходимый спектр сервисных утилит, а также быстро изменять его в случае возникновения такой необходимости. Несмотря на это, существующие преимущества математического пакета Matlab, а именно визуализация результатов исследований, обработка статистических данных и интерпретация математических формул, могут быть использованы при построении универсального программного средства.

4.2 Требования к программному средству

На основе выполненного обзора существующих программных пакетов, были сформированы следующие требования к разрабатываемому программному средству:

- Удобный графический интерфейс.

- Платформенно-независимая библиотека алгоритмов глобальной оптимизации.
- Модульность.
- Объектно-ориентированная реализация.
- Возможность переноса на другую платформу.

4.3 Выбор аппаратных и программных средств

Системные требования:

1. Windows 7 и выше.
2. 512 Мбайт ОЗУ.
3. Процессор семейства Intel и AMD.
4. Дискретная/ встроенная видеокарта с 1 Гбайт DDR3.
5. 10 Мбайт пространства на HDD.

Программные инструменты:

1. Язык реализации: C#.
2. Версия компилятора: MSVC 11.
3. Система сборки: MS Build.
4. IDE: Microsoft Visual Studio 2017.
5. Фреймворк: .NET Framework 4.6.

Дополнительные программные инструменты:

1. Matlab библиотека для C#.

4.4 Обоснование выбора языковой среды

В качестве языковой среды реализации был выбран язык C#. Такой выбор был определен возможностями объектно-ориентированного подхода, заложенными в этом языке, а также хорошей его расширяемостью.

4.5 Функциональные возможности программного средства

Разработанное программное средство поддерживает следующие режимы работы:

1. Решение поставленной задачи глобальной оптимизации одним из алгоритмов.

2. Исследование алгоритмов глобальной оптимизации на решаемой задаче.

4.5.1 Решение поставленной задачи глобальной оптимизации

Перед запуском алгоритма исследователь задаёт стартовую точку и начальные значения области варьирования по каждой координате, а также дополнительные параметры (в зависимости от выбранного алгоритма). По завершению работы программа выдает найденное решение:

- точка оптимума;
- значений функции качества в найденной точке.

Визуальный контроль над ходом работы алгоритма включает:

- изменение значений по каждой координате;
- изменение значений функции качества на каждом шаге работы алгоритма.

4.6 Структура программного средства

Программное средство представляет собой оконное приложение, написанное на языке C# с использованием библиотек MATLAB для визуального представления графиков и располагающихся на них точек и их движения. Главное окно вмещает множество полей для ввода параметров, сгруппированных в соответствии с тем к какой группе параметров они относятся, при этом в самом окне имеется 2 большие вкладки для каждого алгоритма отдельно, рисунок 5, рисунок 6.

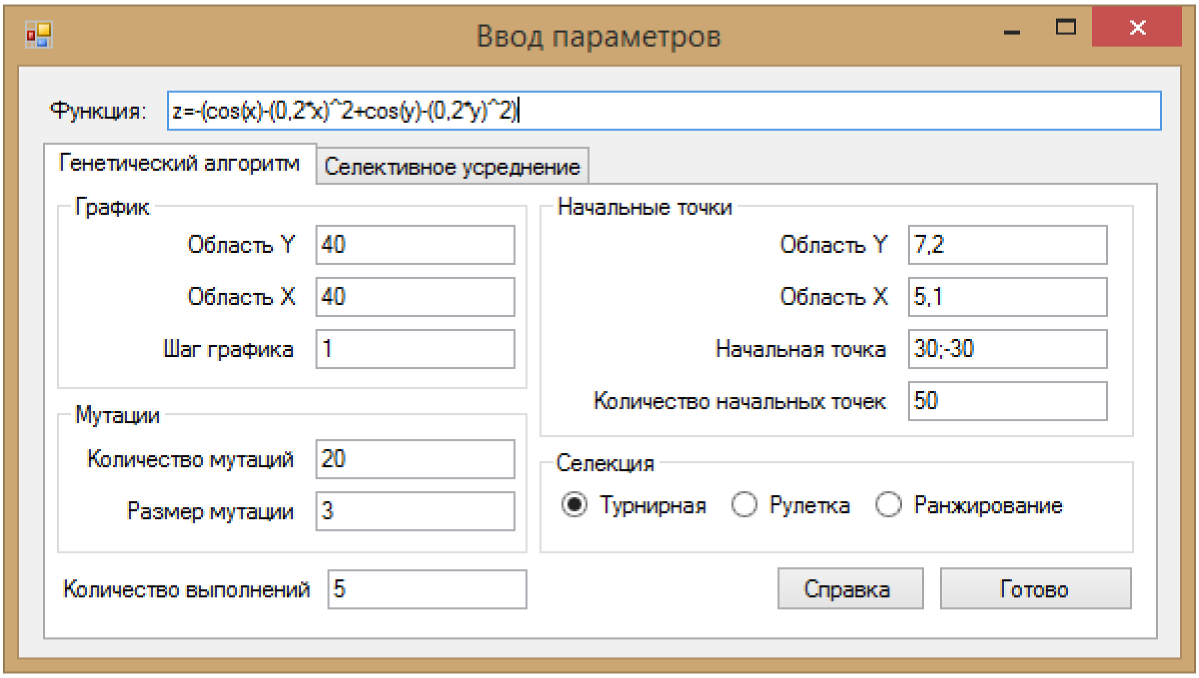

Рисунок 5– Интерфейс главного окна, вкладка – Генетический алгоритм

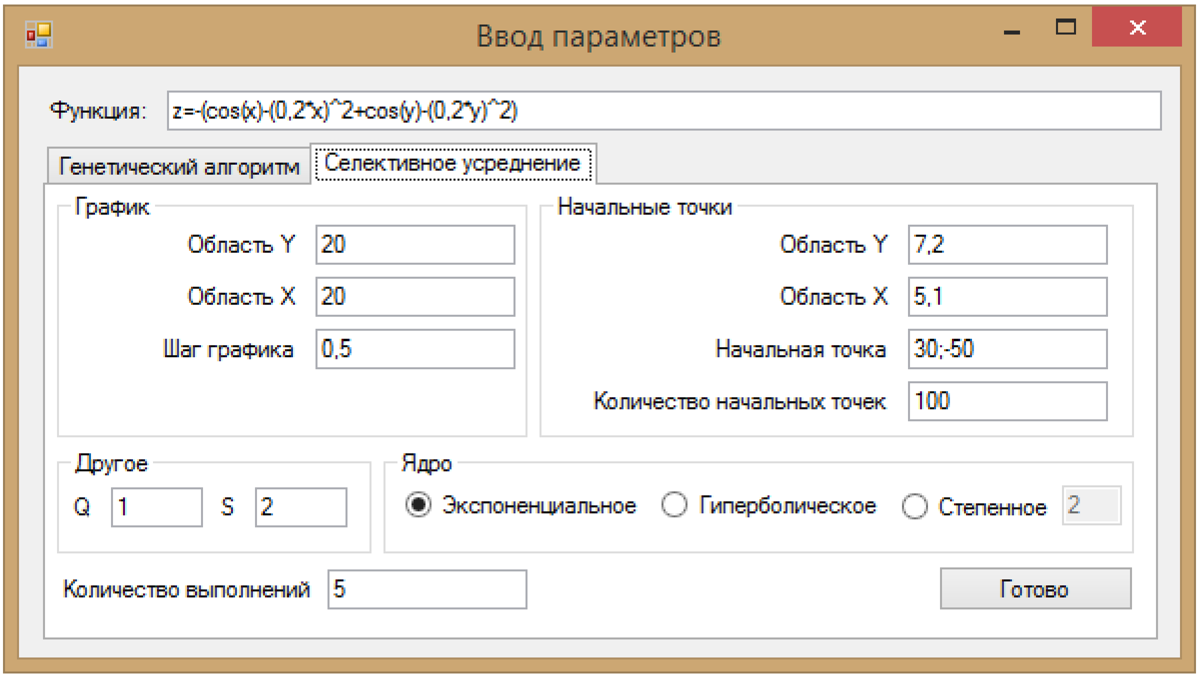

Рисунок 6 – Интерфейс главного окна, вкладка – Селективное усреднение

Второе окно является одним из результатов работы программного средства, служит для вывода информации о проделанной алгоритмом работе. Выглядит одинаково для обоих реализаций алгоритмов, в нём находится элемент хранящий список совершенных алгоритмом действий, набор кнопок “Старт”, “Пауза” и “Стоп”, служащих для запуска, приостановки и полной остановки алгоритма соответственно, а также кнопка вызова журнала проведённых алгоритмом действий являющегося файлом для долгосрочного хранения того же списка, что выводится в описанном выше элементе окна, рисунок 7.

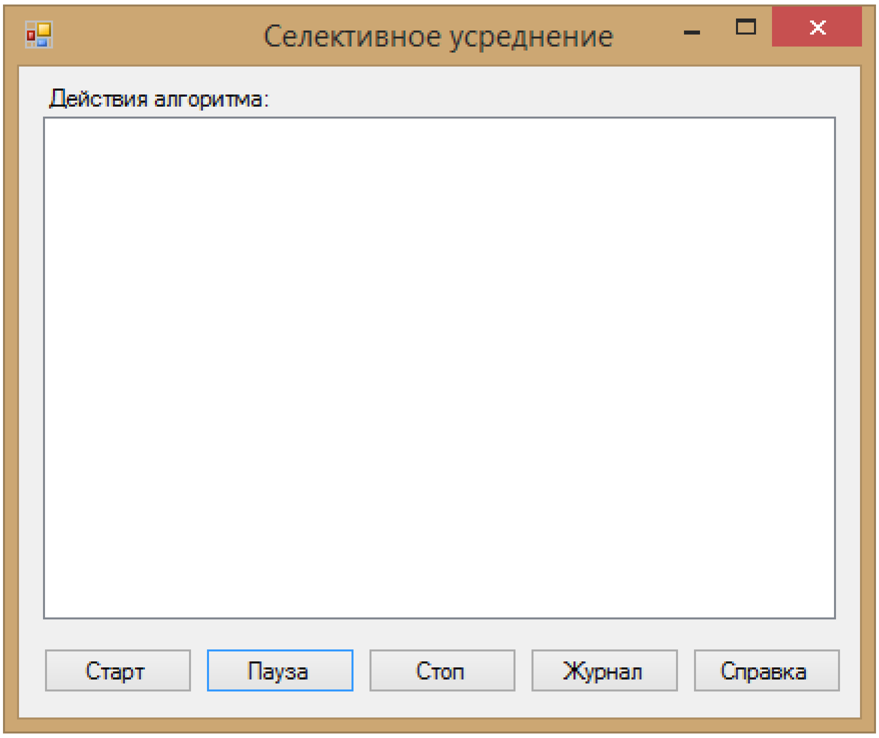

Рисунок 7 – Интерфейс окна вывода на примере Селективного усреднения

Так же результатом работы программного средства является окно отображающее график вводимой функции вместе с отмеченными на нём точками в соответствии с теми, с которыми работает выбранный алгоритм.

Данное окно вызывается при помощи обращения программного средства к программному пакету MATLAB и обладает набором функций, соответствующих окнам, отображающим графиков в данном программном пакете, рисунок 8.

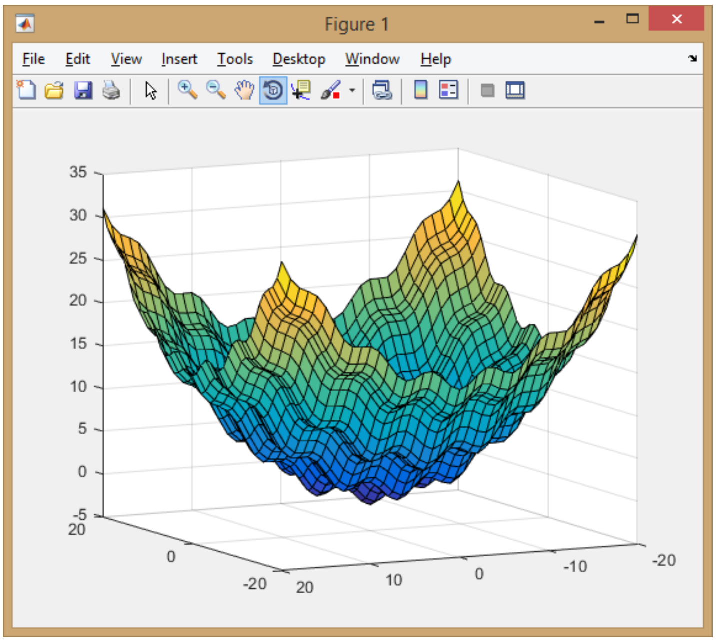

Рисунок 8 – Окно отображающее график введённой функции

5. Численные исследования

Поскольку в качестве результата программное средство выдаёт данные для исследования свойств и анализа алгоритмов далее будут представлены численные исследования данных алгоритмов при помощи разработанного и реализованного средства.

5.1 Тестовая функция №1

Тестовая целевая функция имеет следующий вид

$$f(x_1, x_2) = \cos(x_1) - (0.2 * x_1)^2 + \cos(x_2) - (0.2 * x_2)^2$$

Глобальный минимум (максимум) приходится на точку с координатами $x^* = (0; 0)$: $f(0; 0) = 0$. На рисунке 9 приведен пространственный вид целевой функции.

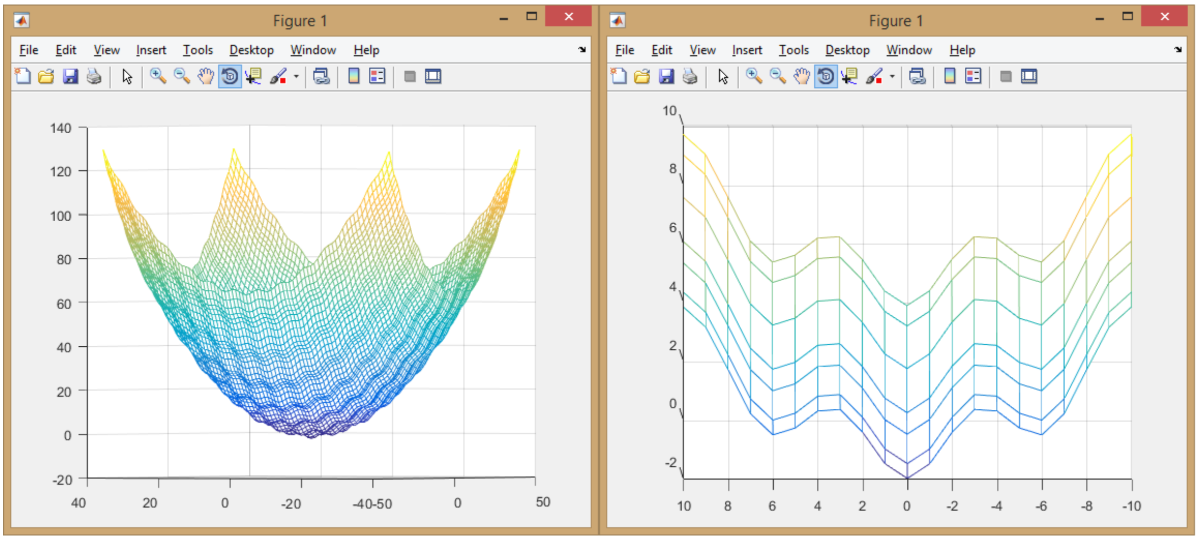

Рисунок 9 – Окно отображающее график введённой функции

Генетический алгоритм.

Решается задача минимизации заданной целевой функции. Параметры алгоритма следующие: начальная точка старта приходится на точку с координатами $x^0 = (30; -30)$, исходная область варьирования $\Delta x^0 = (40; 40)$. Объем выборки пробных точек равен 50. В популяции происходит мутация 20 случайных точек с максимально возможным размером мутации по обоим координатам в 2 единицы. Тип селекции: турнирная.

В ходе работы популяция начинала движение к экстремуму в углу отображаемой поверхности, каждая точка отображалась синим перекрестьем.

С продвижением популяции к экстремуму движение популяции замедлялось, и в среднем при 5 запусках с одинаковыми параметрами потребовалось 30 итераций для нахождения глобального максимума при входных параметрах описанных выше. При повторных запусках в количестве 15 раз были получены те же результаты по количеству требуемых итераций.

На рисунке 10 на фоне пространственного вида целевой функции приведены шаги алгоритма. На рисунке 11 приведены результаты поиска глобального экстремума, выдаваемые программным средством.

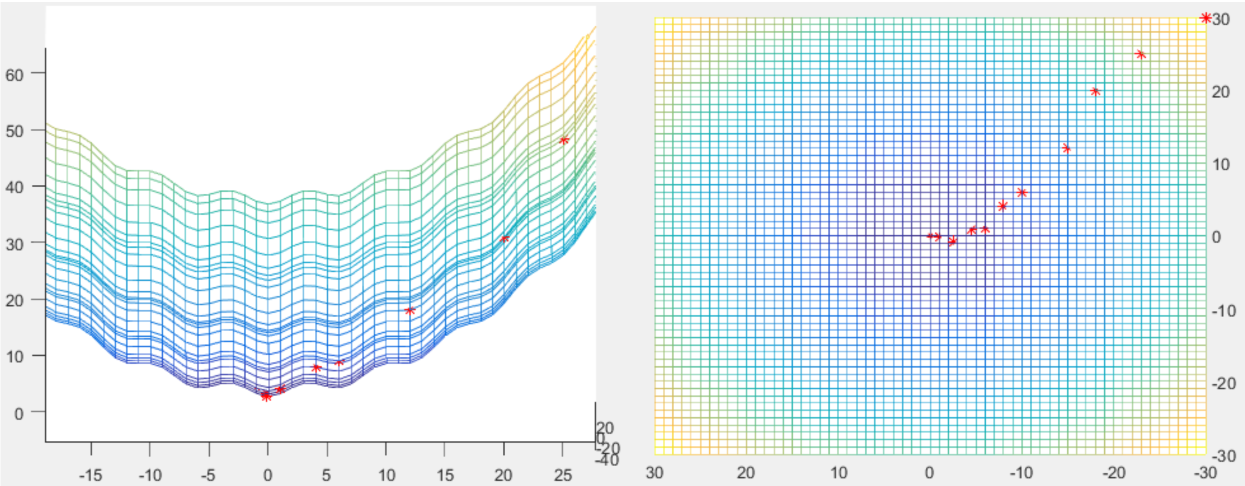

Рисунок 10 – Графический результат работы примера

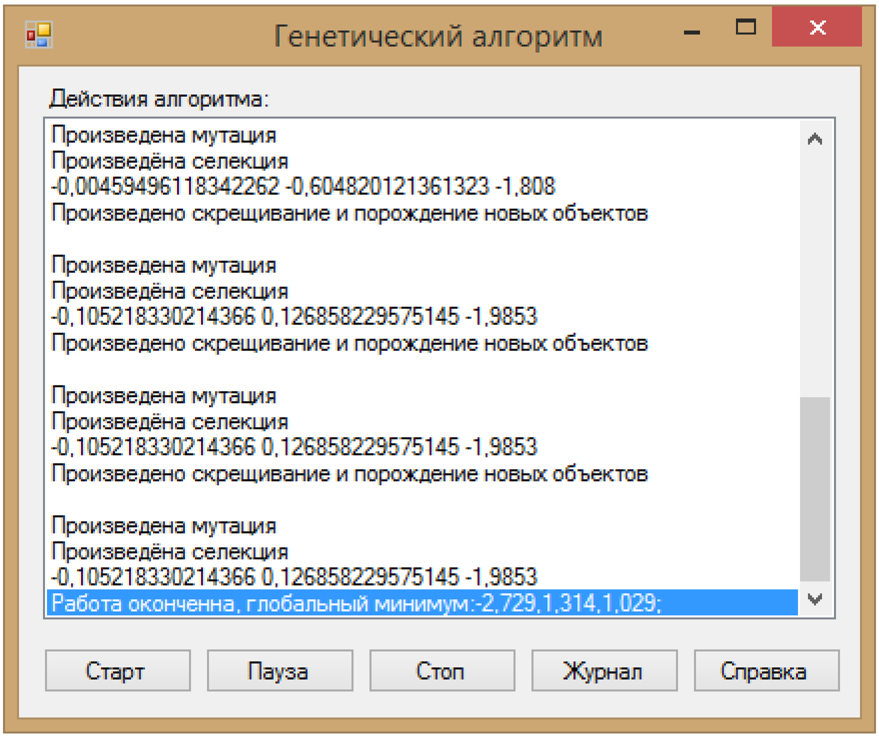

Рисунок 11 – Текстовый результат работы примера

Алгоритм, основанный на методе селективного усреднения искомых координат.

Решается задача минимизации заданной целевой функции. Параметры алгоритма следующие: начальная точка старта приходится на точку с координатами $x^0 = (30; -30)$, исходная область варьирования $\Delta x^0 = (40; 40)$. Объем выборки пробных точек равен 50. Ядро параболическое со степенью селективности $s = 50$. Дополнительные настраиваемые параметры: $q = 2, \gamma_1 = 1$.

На первых шагах алгоритма происходит выход в окрестность глобального минимума, а на остальных шагах осуществляется его уточнение. На рисунке 12 на фоне пространственного вида целевой функции приведены шаги алгоритма. На рисунке 13 приведены результаты поиска глобального экстремума, выдаваемые программным средством.

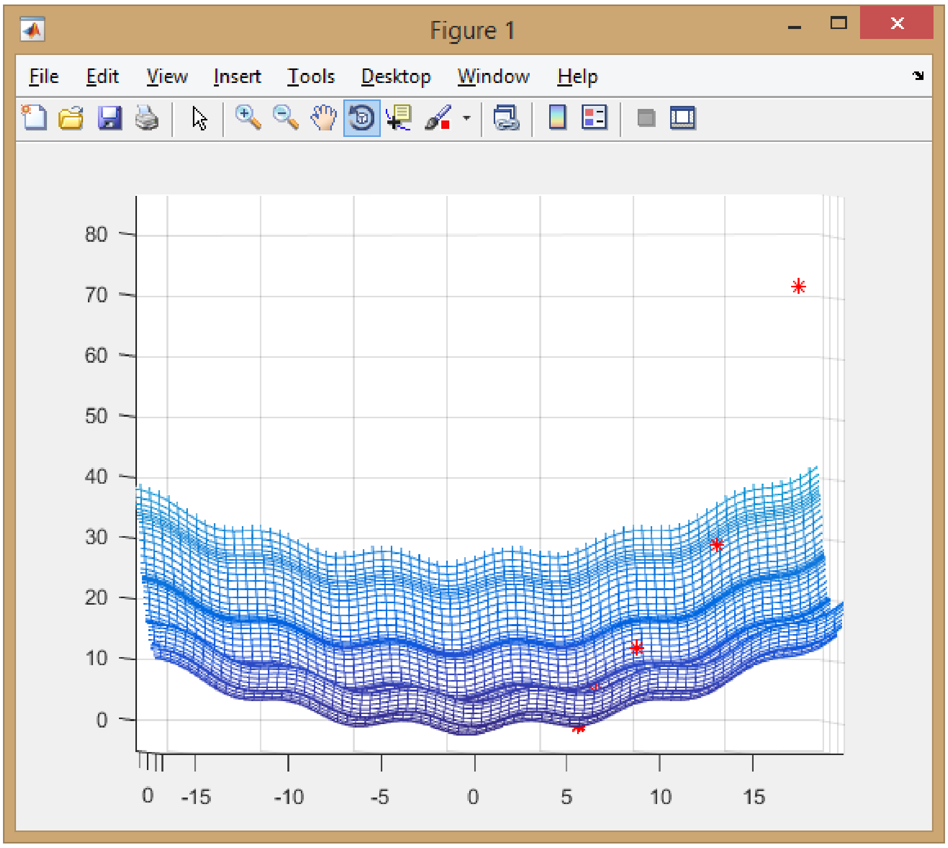

Рисунок 12 – Графический результат работы примера

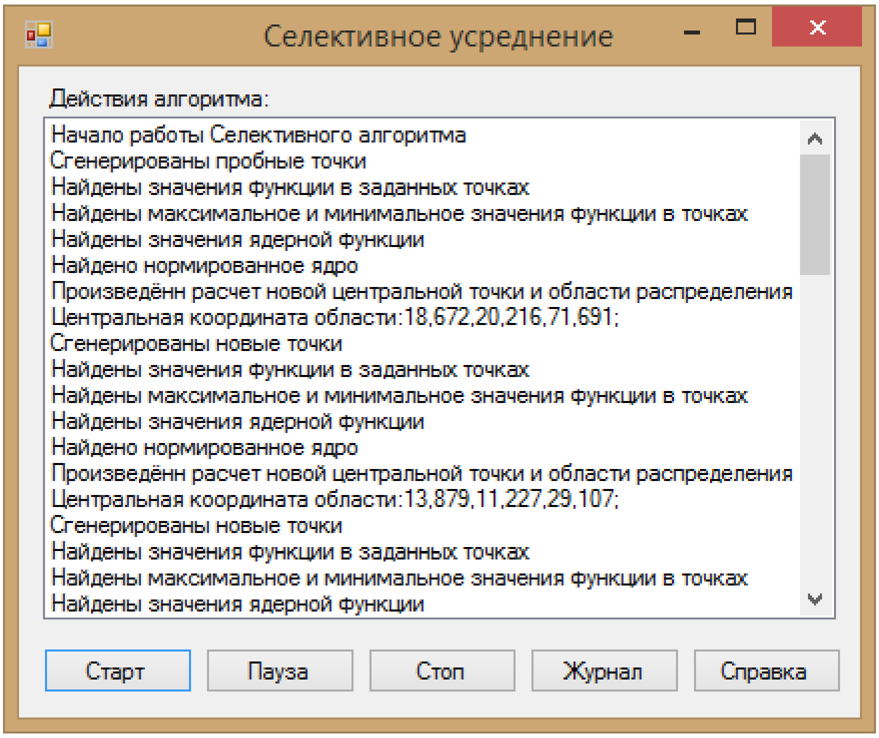

Рисунок 13 – Текстовый результат работы примера

В сравнении с генетическим алгоритмом, алгоритм на основе селективного усреднения искомых переменных потребовал меньше итераций и времени: генетический алгоритм потребовал на выполнение 5 секунд, алгоритм селективного усреднения потребовал менее 3 секунд.

5.2 Тестовая функция №2

Тестовая целевая функция имеет следующий вид

$$z_1(x) = 6|x_1|^2 + 7|x_2|^2$$

$$z_2(x) = 5|x_1 + 2|^{0.5} + 5|x_2|^{0.5} + 6$$

$$z_3(x) = 5|x_1|^{1.3} + 5|x_2 + 2|^{1.3} + 5$$

$$z_4(x) = 4|x_1|^{0.8} + 3|x_2 - 4|^{1.2} + 8$$

$$z_5(x) = 6|x_1 - 2|^{1.1} + 4|x_2 - 2|^{1.7} + 7$$

$$z_6(x) = 5|x_1 - 4|^{1.1} + 5|x_2|^{1.8} + 9$$

$$z_7(x) = 6|x_1 - 4|^{0.6} + 7|x_2 - 4|^{0.6} + 4$$

$$z_8(x) = 6|x_1 + 4|^{0.6} + 6|x_2 - 4|^{1.6} + 3$$

$$z_9(x) = 3|x_1 + 4|^{1.2} + 3|x_2 + 4|^{0.5} + 7.5$$

$$z_{10}(x) = 2|x_1 - 3|^{0.9} + 4|x_2 + 5|^{0.3} + 8.5$$

$$\boldsymbol{f(x_1, x_2) = \min(z_i(x), i = \overline{1, 10}\}}$$

Приведенная многоэкстремальная функция имеет 10 минимумов. Глобальный минимум приходится на точку с координатами $x^* = (0; 0)$: $f(0; 0) = 0$. На рисунке 14 приведен пространственный вид целевой функции.

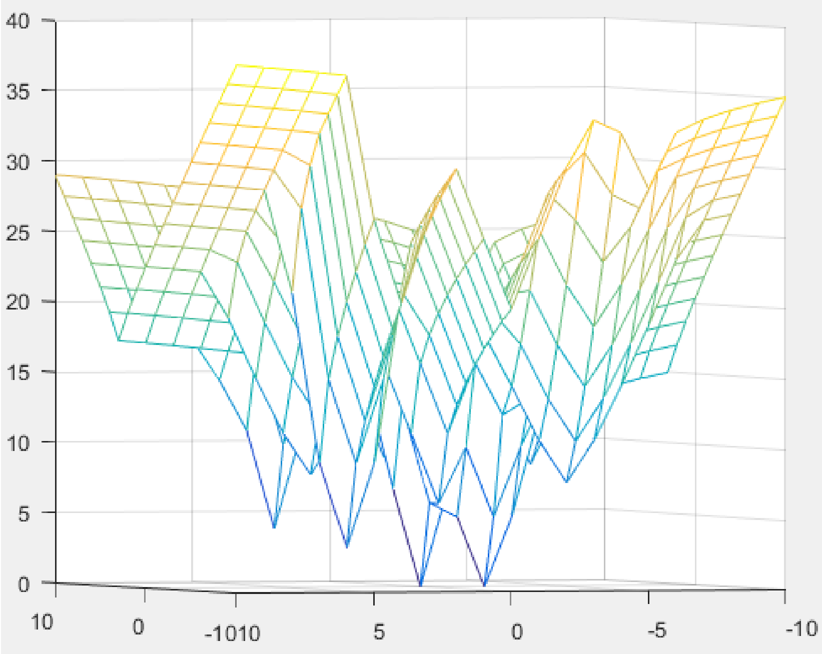

Рисунок 14 – пространственный вид целевой функции

Генетический алгоритм.

Решается задача максимизации целевой функции (5.2.1). Параметры алгоритма следующие: начальная точка старта приходится на точку с координатами $x^0 = (5; 5)$, исходная область варьирования $\Delta x^0 = (10; 10)$. Объем выборки пробных точек равен 50. В популяции происходит мутация 30 случайных точек с максимально возможным размером мутации по обоим координатам в 3 единиц. Тип селекции: турнирная.

В ходе работы популяция начинала движение к экстремуму в углу отображаемой поверхности, каждая точка отображалась звездой (рисунок 15). С продвижением популяции к экстремуму движение популяции замедлялось, и в среднем при 5 запусках с одинаковыми параметрами потребовалось 20 итераций для нахождения глобального максимума при входных параметрах описанных выше. При повторных запусках в количестве 15 раз были получены те же результаты по количеству требуемых итераций. На рисунке 16 на фоне пространственного вида целевой функции приведены шаги алгоритма. На рисунке 17 приведены результаты поиска глобального экстремума, выдаваемые программным средством.

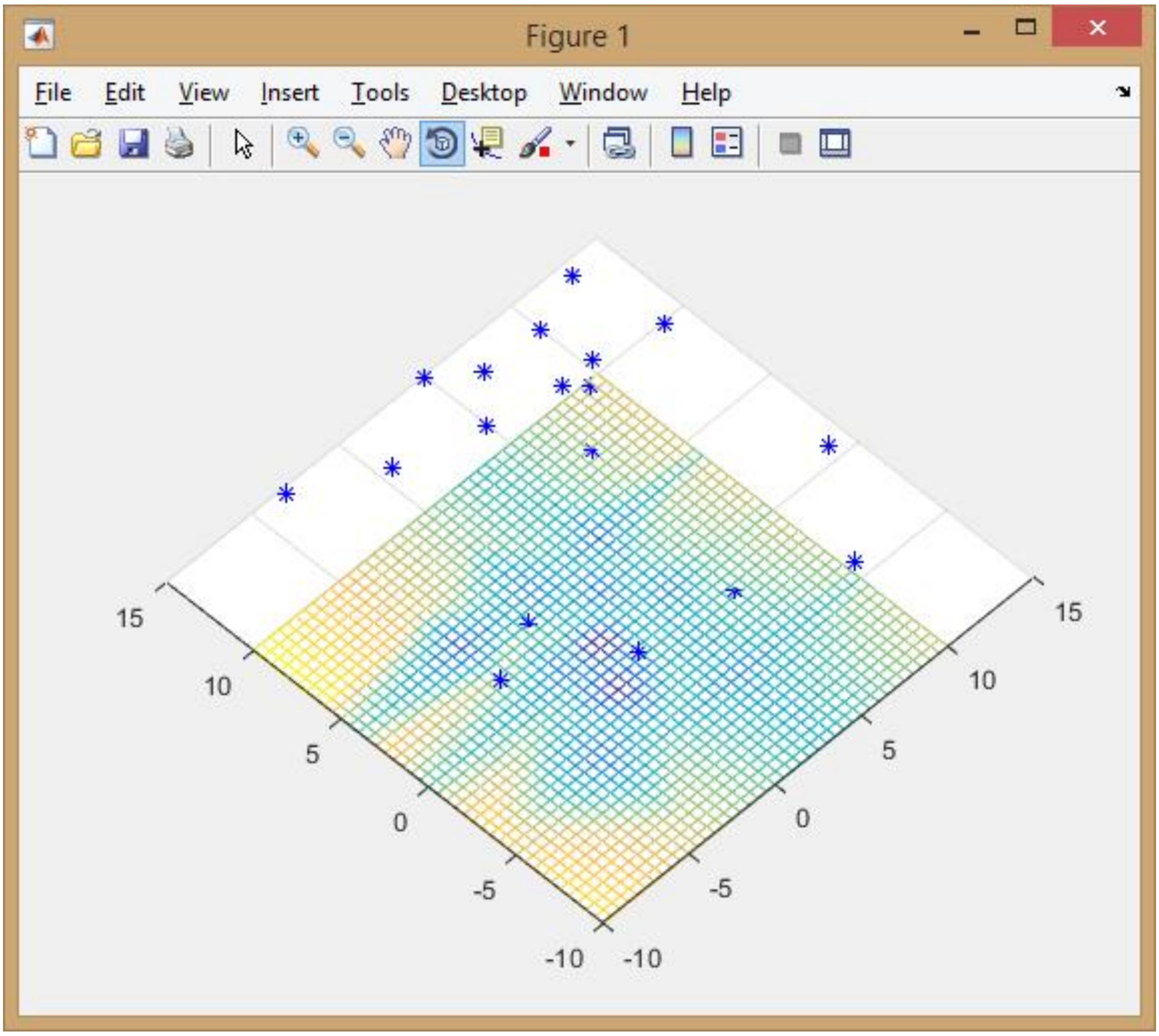

Рисунок 15 – Начало работы алгоритма

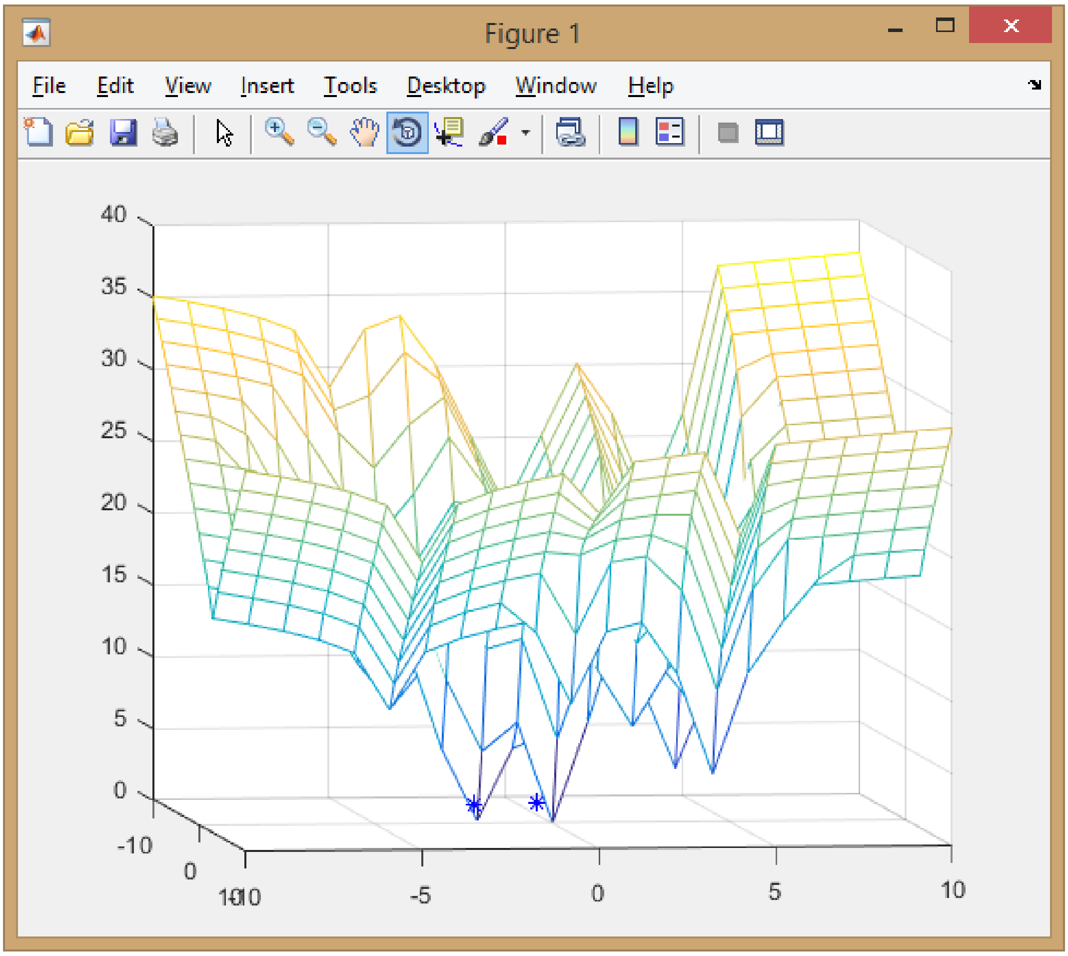

Рисунок 16 –Графический результат работы алгоритма

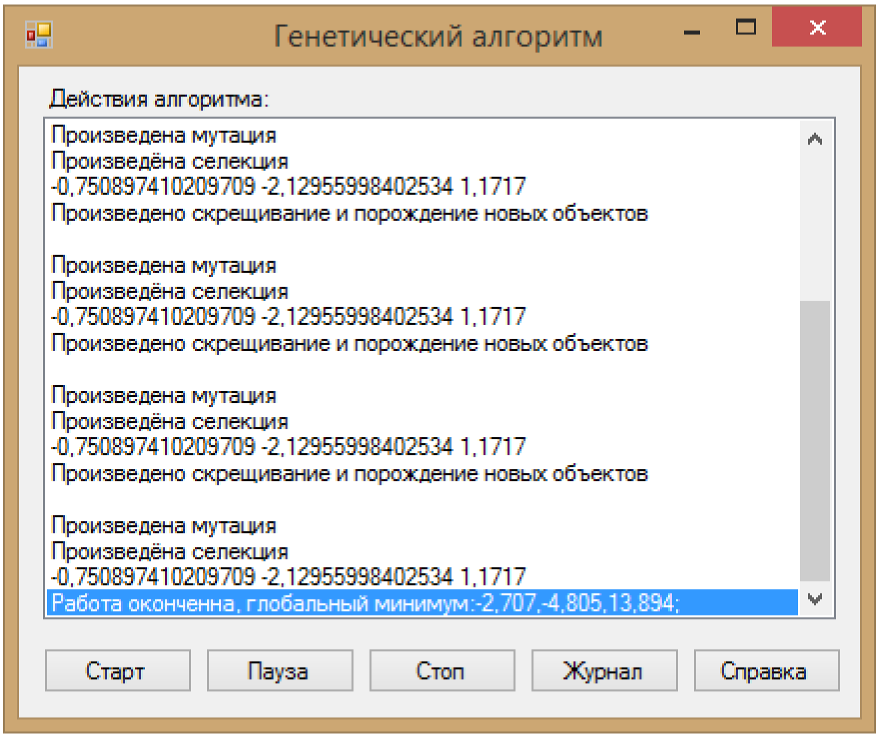

Рисунок 17 – Текстовый результат работы алгоритма

Алгоритм, основанный на методе селективного усреднения искомых координат.

Решается задача минимизации заданной целевой функции. Параметры алгоритма следующие: начальная точка старта приходится на точку с координатами $x^0 = (5; 5)$, исходная область варьирования $\Delta x^0 = (10; 10)$. Объем выборки пробных точек равен 50. Ядро параболическое со степенью селективности $s = 50$. Дополнительные настраиваемые параметры: $q = 2, \gamma_1 = 1$.

На первых шагах алгоритма происходит выход в окрестность глобального минимума, а на остальных шагах осуществляется его уточнение. На рисунке 18 на графике пространственного вида целевой функции приведены шаги алгоритма. На рисунке 19 приведены результаты поиска глобального экстремума, выдаваемые программным средством.

В сравнении с генетическим алгоритмом, алгоритм на основе селективного усреднения искомых переменных потребовал меньше итераций и времени: генетический алгоритм потребовал от 6 до 7 секунд на выполнение, алгоритм селективного усреднения справился менее чем за 3 секунды.

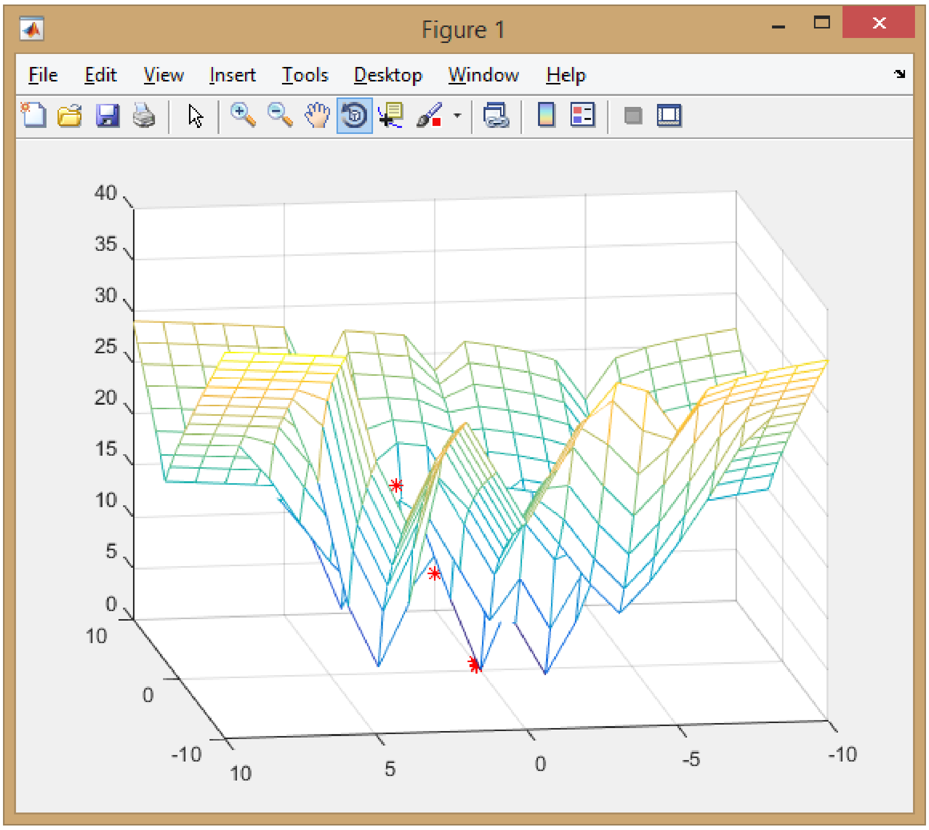

Рисунок 18 – Графический результат работы алгоритма

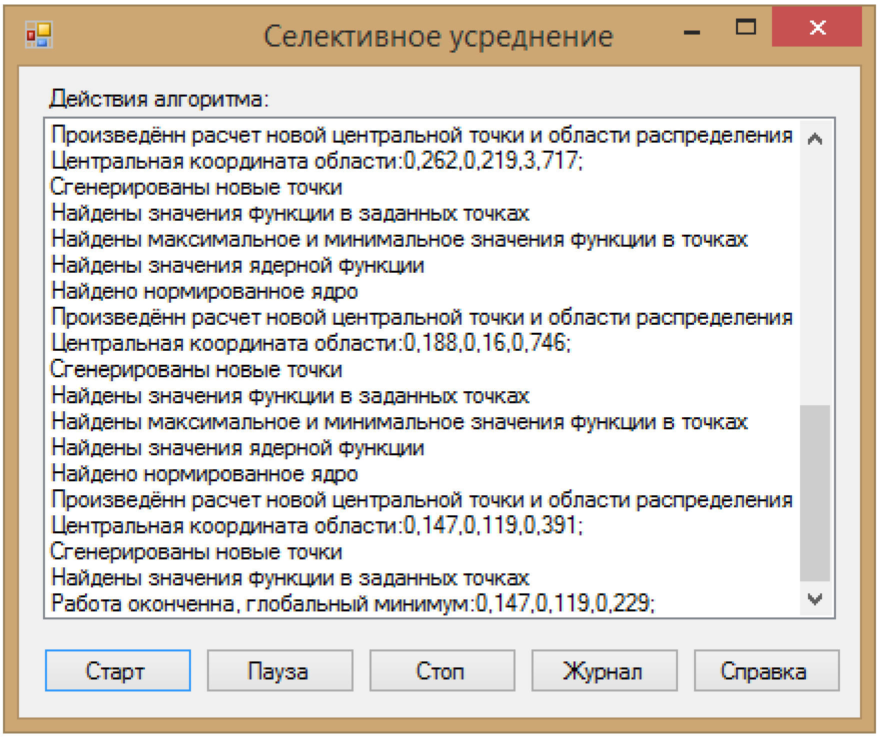

Рисунок 19 – Текстовый результат работы алгоритма

ЗАКЛЮЧЕНИЕ

Цель ВКР была достигнута – разработано универсальное программное средство для исследования свойств алгоритмов глобальной оптимизации, показана на практике возможность использования данного программного средства для исследовательских целей. Изучена предметная область. Получены теоретические знания, а так же практические навыки по реализации и работе с алгоритмами глобальной оптимизации. Программа разработана для использования в исследовательских целях.

СПИСОК ИСПОЛЬЗОВАННЫЙ ИСТОЧНИКОВ

1. Стронгин Р. Г. Поиск глобального оптимума. – М.: Знание, 1990. – 48 с

2. Жиглявский А. А., Жилинскас А. Г. Методы поиска глобального оптимума. – М.: Наука, 1991. – 247 с.

3. Horst R., Tuy H. Global optimization – Deterministic approaches. Third Edition. – Berlin: Springer, 1996. – 729 p

4. Zhigljavsky A., Zilinskas A. Stochastic global optimization. – Berlin: Springer, 2008. – 269 p

5. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учебное пособие. - М.: Физматлит, 2006. – 320 с.

6. Рубан А. И. Глобальная оптимизация методом усреднения координат: Монография. – Красноярск: ИПЦ КГТУ, 2004. – 303 с.